



Mengenal dan Mengoptimalkan Fitur Macro Excel

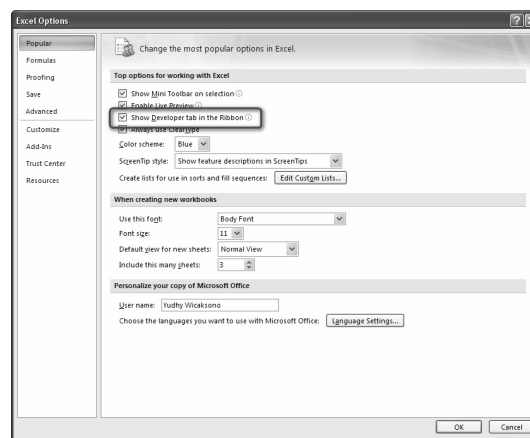
Macro, atau biasa juga dikenal dengan istilah Visual Basic for Application (VBA), merupakan pengembangan bahasa pemrograman Visual Basic yang diterapkan dalam aplikasi Excel. Berbeda dengan program pengembang Visual Basic, pemrograman yang dibuat menggunakan Macro hanya dapat dibangun dan digunakan pada aplikasi Excel. Program yang dibuat menggunakan Macro tidak dapat berjalan, sebelum Anda menjalankan Excel terlebih dahulu. Pemrograman Macro Excel mempunyai beberapa keuntungan sebagai berikut:

- © **Menghemat waktu.** Penyelesaian pekerjaan menggunakan Macro lebih cepat dibandingkan dengan cara manual, karena prosesnya dikerjakan secara otomatis.
- © **Menghemat tenaga.** Selain menghemat waktu, penyelesaian pekerjaan menggunakan Macro juga dapat menghemat tenaga.
- © **Mengurangi tingkat kesalahan.** Kemungkinan adanya kesalahan dalam menyelesaikan pekerjaan secara manual dapat saja terjadi, meskipun Anda seorang yang sangat ahli dalam menggunakan Excel. Penyelesaian pekerjaan menggunakan Macro secara konsisten akan menyelesaikan suatu pekerjaan berdasarkan perintah yang tertulis dalam kode Macro sehingga tingkat kesalahan yang mungkin timbul sangat kecil. Kesalahan hanya dapat terjadi jika ada kesalahan perintah pada kode Macro.

1.1 PENGGUNAAN FITUR MACRO

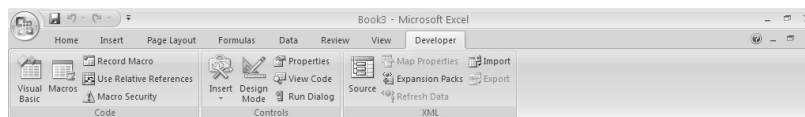
Sebelum menggunakan fitur Macro, Anda harus menampilkan tab Developer dalam Ribbon terlebih dahulu. Berikut langkah-langkah untuk mengaktifkan tab Developer:

1. Untuk pengguna Excel 2007, klik **Office Button** kemudian pilih **Excel Options**. Muncul kotak dialog Excel Options.



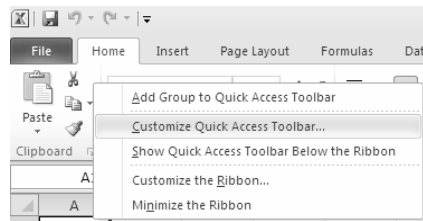
Gambar 1.1 Menampilkan tab Developer Excel 2007.

2. Pilih opsi **Popular**. Beri tanda centang pada pilihan **Show Developer tab in the Ribbon** lalu klik tombol **OK**. Tampilan tab Developer dalam Ribbon terlihat seperti pada Gambar 1.2.



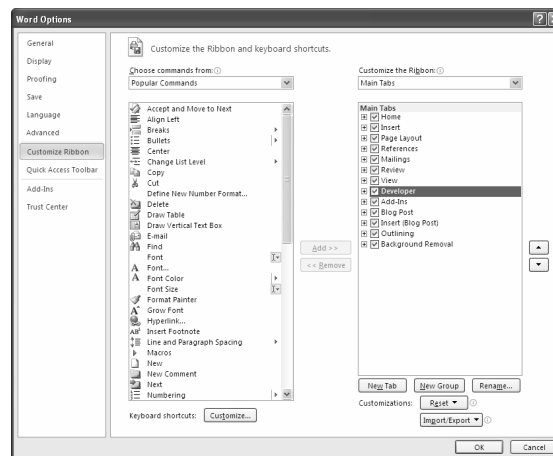
Gambar 1.2 Tab Developer dalam Ribbon.

3. Untuk menampilkan tab Developer pada Ribbon Excel 2010, klik kanan area Ribbon kemudian pilih menu **Customize the Ribbon...** Muncul kotak dialog Excel Options pada pilihan Customize Ribbon.



Gambar 1.3 Menu klik kanan area Ribbon.

4. Anda juga dapat menampilkan kotak dialog Excel Options dengan cara klik tab **File** kemudian pilih **Options**. Muncul kotak dialog Excel Options. Pilih opsi **Customize Ribbon**.



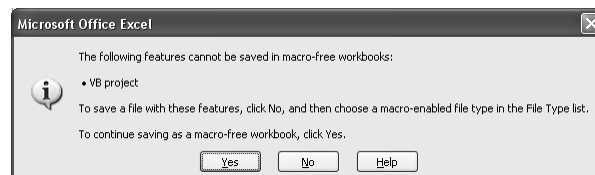
Gambar 1.4 Menampilkan tab Developer Excel 2010.

5. Pilih **Main Tabs** pada kotak pilihan **Customize the Ribbon**: Beri tanda centang tab **Developer** dalam daftar di sebelah kanan. Klik tombol **OK**.

1.2 FORMAT FILE

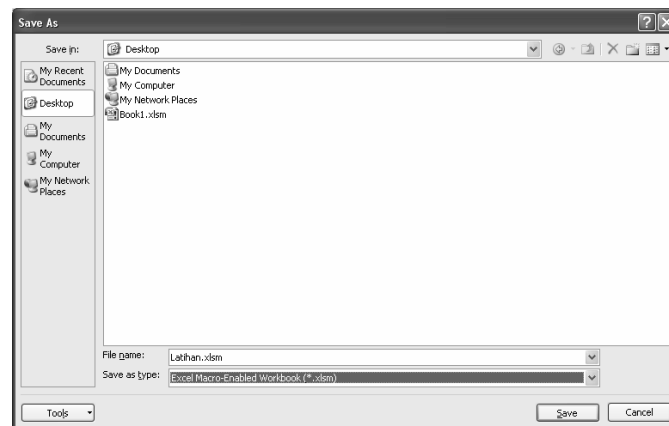
Workbook standar Excel Excel 2007 dan Excel 2010 dengan format XLSX tidak mendukung Macro. Untuk menyimpan workbook Excel 2007 atau Excel 2010 yang mengandung Macro, Anda harus menggunakan format XLSM atau XLS (Excel 97-2003).

1. Klik tombol **Office Button** kemudian pilih menu **Save As** untuk menyimpan workbook. Bagi pengguna Excel 2010 klik tab **File** kemudian pilih menu **Save As**. Muncul kotak dialog Save As. Jika Anda menyimpan workbook yang mengandung Macro dengan format file XLSX, muncul kotak pesan seperti berikut.



Gambar 1.5 Kotak pesan format penyimpanan workbook.

2. Klik tombol **Yes** jika workbook akan tetap disimpan dalam format file XLSX tanpa mengaktifkan Macro. Jika opsi ini yang Anda pilih, Macro dalam workbook akan dihapus.
3. Untuk menyimpan workbook dengan Macro, klik tombol **No**. Muncul kotak dialog Save As. Ketikkan nama file pada kotak isian File name:. Pada kotak pilihan Save as type: pilih format file **Excel Macro-Enabled Workbook (*.xlm)** atau **Excel 97-2003 Workbook (*.xls)** kemudian klik tombol **Save**.

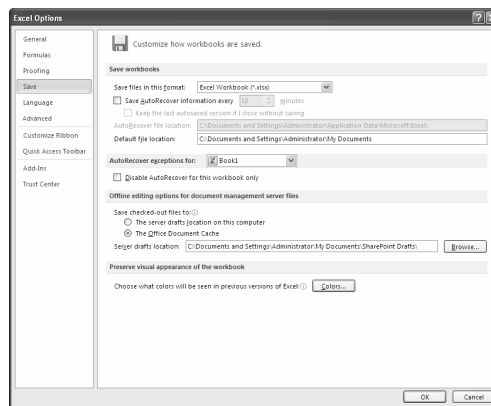


Gambar 1.6 Kotak dialog Save As.

1.3 MENONAKTIFKAN PENYIMPANAN WORKBOOK OTOMATIS

Excel menyediakan fitur yang digunakan untuk menyimpan workbook secara otomatis setiap interval waktu tertentu. Cara ini memang sangat bermanfaat untuk menghindari kejadian yang tidak diharapkan, misalnya listrik yang tiba-tiba padam. Namun demikian, penyimpanan secara otomatis dalam bentuk workbook backup tidak mendukung fitur Macro sehingga terkadang malah menyebabkan error. Oleh karena itu, untuk sementara fitur tersebut kita nonaktifkan.

1. Bagi pengguna Excel 2007, klik **Office Button** lalu pilih **Excel Options**. Untuk pengguna Excel 2010, klik tab **File** kemudian pilih menu **Options**. Muncul kotak dialog Excel Options.
2. Pilih opsi **Save** kemudian hilangkan tanda centang pada pilihan **Save AutoRecover information every**. Jika sudah, klik tombol **OK**.

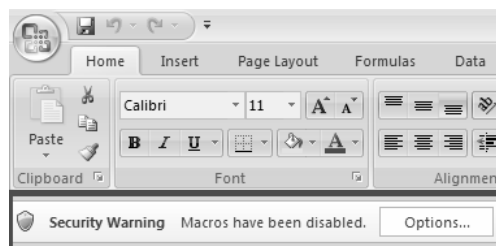


Gambar 1.7 Menonaktifkan penyimpanan workbook otomatis.

1.4 KEAMANAN MACRO

Otomatisasi dalam Macro Excel terkadang disalahgunakan oleh pengguna yang tidak bertanggung jawab untuk menyebarkan kode

Macro yang berpotensi menimbulkan kerugian, misalnya kode Macro untuk menghapus data. Untuk mengantisipasi hal tersebut, Excel secara default akan memberi peringatan jika pengguna membuka workbook yang mengandung Macro. Excel 2007 secara otomatis akan menonaktifkan Macro saat workbook yang mengandung Macro pertama kali dibuka. Muncul peringatan keamanan seperti terlihat pada Gambar 1.8.



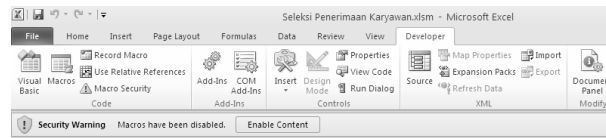
Gambar 1.8 Peringatan keamanan Macro Excel 2007.

Klik tombol **Options...** Muncul kotak dialog Microsoft Office Security Options. Untuk mengaktifkan Macro pilih opsi **Enable this content** kemudian klik tombol **OK**.



Gambar 1.9 Kotak dialog Microsoft Office Security Options.

Excel 2010 secara otomatis akan menonaktifkan Macro ketika workbook yang mengandung Macro dibuka. Muncul peringatan keamanan seperti terlihat pada Gambar 1.10. Klik tombol **Enable Content** untuk mengaktifkan Macro.



Gambar 1.10 Peringatan keamanan Macro Excel 2010.

Excel 2010 selanjutnya akan menampilkan kotak dialog Microsoft Office Excel Security Notice jika Anda membuka workbook yang mengandung Macro. Apabila Anda tidak mengenal Macro pada workbook tersebut, klik tombol **Disable Macros** untuk menon-aktifkan Macro. Untuk mengaktifkan Macro, klik tombol **Enable Macros**.

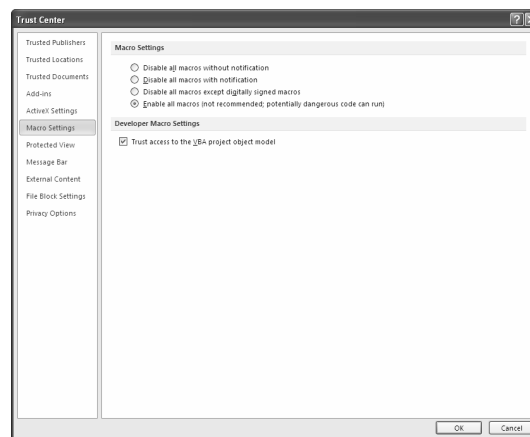


Gambar 1.11 Kotak dialog Microsoft Office Excel Security Notice.

Untuk mengatur keamanan Macro, klik tombol **Macro Security** dalam tab Developer group Code. Muncul kotak dialog Trust Center pada pilihan **Macro Settings**. Anda dapat melakukan pengaturan keamanan Macro sebagai berikut:

- ⦿ **Disable all macros without notification.** Apabila opsi ini dipilih, Macro dalam workbook tidak akan dijalankan ketika dibuka.
- ⦿ **Disable all macros except digitally signed macros.** Jika opsi ini dipilih, Macro dalam workbook tidak akan dijalankan ketika dibuka, kecuali untuk Macro yang dikembangkan oleh pengembang terpercaya.
- ⦿ **Disable all macros with notification.** Apabila opsi ini dipilih, Excel akan menampilkan informasi peringatan ketika Anda membuka workbook yang mengandung Macro. Dijalankan atau tidaknya Macro selanjutnya tergantung pada pilihan Anda.

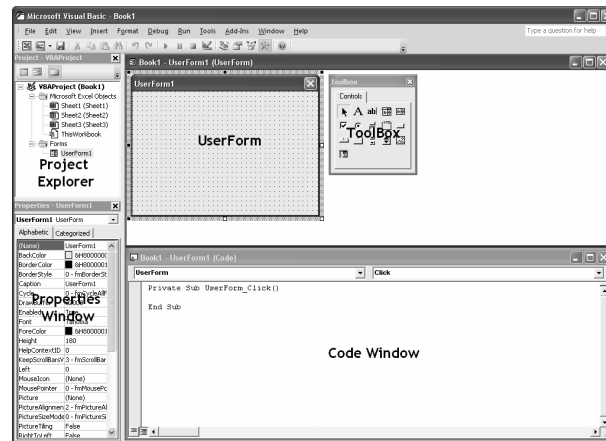
- © **Enabled all macros (not recommended; potentially dangerous code can run).** Apabila opsi ini dipilih, Macro yang ada dalam workbook akan selalu dijalankan, tanpa melalui informasi peringatan. Opsi ini tidak direkomendasikan, terutama untuk macro dalam workbook yang tidak Anda kenal, karena berpotensi menimbulkan kerugian. Jika Anda tetap memilih opsi ini, beri tanda centang pada pilihan **Trust access to the VBA project object model**.



Gambar 1.12 Kotak dialog Trust Center.

1.5 VISUAL BASIC EDITOR

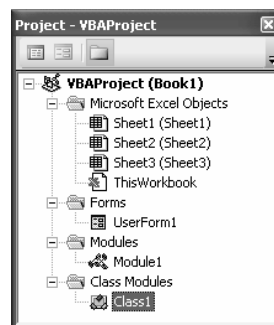
Visual Basic Editor merupakan lingkungan kerja, tempat di mana Macro Excel dibuat. Tampilan Visual Basic Editor sangat berbeda dengan tampilan utama Excel. Untuk menampilkan Visual Basic Editor, klik tombol **Visual Basic** dalam tab Developer group Code. Visual Basic Editor juga dapat ditampilkan menggunakan kombinasi tombol **Alt+F11** pada keyboard.




Gambar 1.13 Tampilan Visual Basic Editor.

1.5.1 Project Explorer

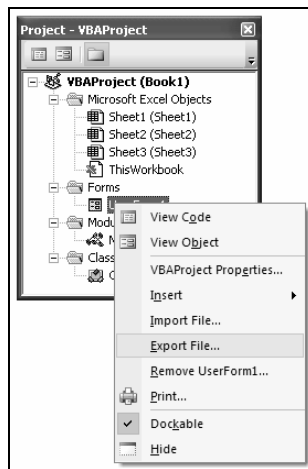
Project Explorer digunakan untuk melakukan navigasi terhadap seluruh objek yang ada dalam proyek VBA sebuah workbook. Secara garis besar, objek dikelompokkan ke dalam **Microsoft Excel Objects**, **Forms**, **Modules** dan **Class Modules**.



Gambar 1.14 Tampilan Project Explorer.

Untuk menampilkan atau mengaktifkan Project Explorer, pilih menu **View > Project Explorer** (atau tekan kombinasi **Ctrl+R**). Anda juga dapat menampilkan Project Explorer dengan cara klik tombol **Project Explorer**  yang terdapat dalam toolbar Standard.

Melalui Project Explorer, Anda dapat menampilkan kode Macro, menampilkan objek, mengatur properti VBA, menambahkan objek, mencetak objek dan kode yang terdapat di dalamnya serta melakukan impor dan ekspor objek. Klik kanan Project Explorer pada objek yang Anda inginkan kemudian muncul daftar menu seperti terlihat pada Gambar 1.15.

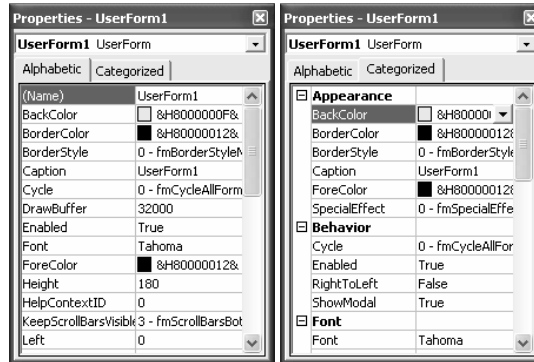


Gambar 1.15 Daftar menu klik kanan Project Explorer.

1.5.2 Window Properties

Window Properties digunakan untuk menampilkan properti yang dimiliki objek. Untuk menampilkan atau mengaktifkan window Properties, pilih menu **View > Properties Window** atau klik tombol **Properties Window** (🔍) pada toolbar Standard. Cara yang sama juga dapat dilakukan dengan menekan tombol **F4** pada keyboard.

Properti objek pada window Properties dapat dilihat berdasarkan urutan abjad (**Alphabetic**) ataupun berdasarkan kategori (**Categorized**). Window Properties secara otomatis akan menampilkan properti objek yang sedang aktif (terpilih).



Gambar 1.16 Tampilan window Properties.

1.5.3 Window Code

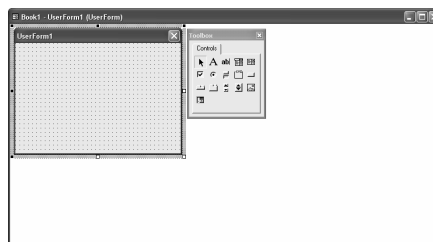
Window Code digunakan untuk melihat, membuat atau melakukan modifikasi kode Macro. Pada window Code, terdapat kotak pilihan Object Selector dan Event Selector. Kotak pilihan Object Selector digunakan untuk memilih objek atau koleksi objek yang akan ditampilkan kode Macro-nya, sedangkan kotak pilihan Event Selector digunakan untuk memilih event pada objek terpilih.



Gambar 1.17 Tampilan window Code.


1.5.4 Window Object

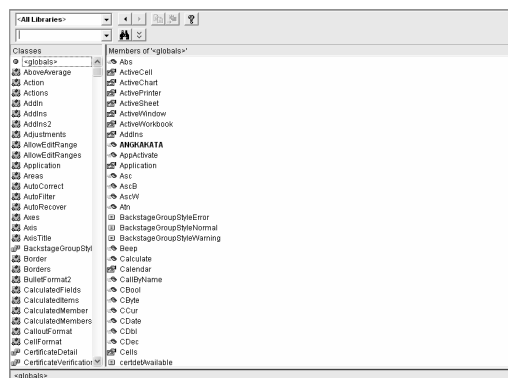
Window Object merupakan tempat yang digunakan untuk menampilkan, membuat dan mengatur objek UserForm serta objek kontrol dalam UserForm secara visual. Untuk menampilkan window Object, pilih menu **View > Object**. Anda juga dapat menampilkan window Object dengan menekan kombinasi tombol **Shift+F7** pada keyboard.



Gambar 1.18 Tampilan window Object.

1.5.5 Object Browser

Object Browser digunakan untuk menampilkan atau melakukan pencarian terhadap semua objek, koleksi objek, properti, method atau event yang terdapat dalam VBA. Untuk menampilkan Object Browser, pilih menu **View > Object Browser**, atau tekan tombol **F2**. Anda juga dapat menampilkan Object Browser dengan cara klik tombol **Object Browser**  dalam toolbar Standard.




Gambar 1.19 Tampilan Object Browser.

Untuk mencari informasi objek, properti, method atau event tertentu menggunakan Object Browser, lakukan langkah-langkah sebagai berikut:

1. Ketikkan objek, properti, method atau event yang akan Anda cari pada kotak pilihan Search Text. Dalam contoh kali ini ketikkan **Italic**, untuk mencari informasi mengenai properti Italic.




Gambar 1.20 Menggunakan fasilitas pencarian.






2. Klik tombol **Search**  atau tekan tombol **Enter** pada keyboard. Hasil pencarian kemudian ditampilkan seperti terlihat pada Gambar 1.21.

Search Results		
Library	Class	Member
Excel	TextEffectFormat	FontItalic
Excel	Font	Italic
Office	Font2	Italic
stdole	StdFont	Italic

Gambar 1.21 Tampilan hasil pencarian.

3. Klik salah satu item hasil pencarian yang Anda inginkan, misalnya pilih properti **Italic** pada Library Excel dan Class Font. Informasi properti yang Anda pilih akan ditampilkan di bagian pojok kiri bawah Object Browser.
4. Untuk mengcopy properti yang Anda pilih, klik tombol **Copy to Clipboard** . Tekan kombinasi **Ctrl+V** untuk menyalin properti yang sudah Anda copy.

Object Browser menggunakan ikon yang berbeda untuk memudahkan Anda dalam membedakan objek, properti, method atau event. Berikut ikon yang digunakan Object Browser untuk membedakan objek, properti, method atau event:

- ⊙  ikon objek.
- ⊙  ikon properti.
- ⊙  ikon event.
- ⊙  ikon method.
- ⊙  ikon predefined constant, yaitu konstanta yang secara default sudah disediakan. Konstanta untuk VBA diawali dengan prefiks vb, misalnya vbYes. Untuk Excel, konstanta diawali dengan prefiks xl, misalnya xlChart.


1.6 KODE MACRO

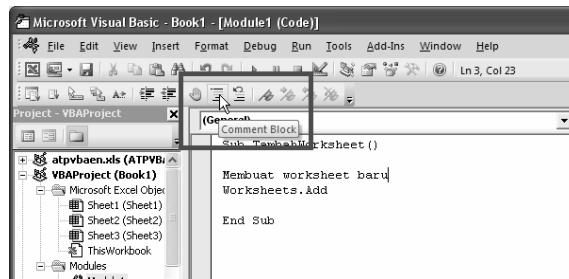
Kode Macro merupakan serangkaian tulisan perintah yang akan dilaksanakan ketika Macro dijalankan. Kode Macro akan mengontrol dan menentukan dijalkannya sebuah Macro. Kode Macro dapat ditulis pada objek workbook, worksheet, Module, UserForm atau Class Module.

1.6.1 Komentar

Anda dapat menambahkan komentar untuk memberi keterangan pada baris kode Macro tertentu. Komentar dapat ditambahkan pada suatu baris dengan menuliskan tanda petik satu (') di depan statement yang ingin dinyatakan sebagai komentar. Komentar tidak dianggap sebagai perintah sehingga tidak akan dijalankan. Perhatikan contoh berikut ini:

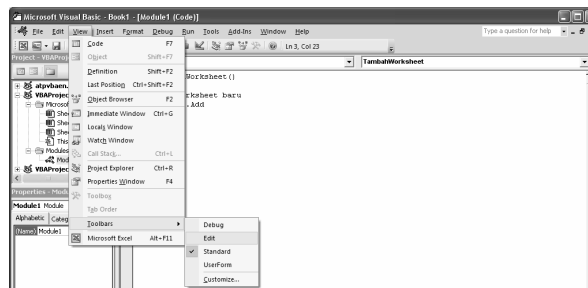
```
'Membuat worksheet baru
Worksheets.Add
```

'Membuat worksheet baru dianggap bukan perintah melainkan komentar sehingga tidak akan dijalankan. `Worksheets.Add` dianggap sebagai perintah sehingga akan dijalankan. Untuk membuat komentar dengan mudah, pilih atau blok baris kode yang akan dibuat menjadi komentar kemudian klik ikon **Comment Block**  pada toolbar Edit.



Gambar 1.22 Menambahkan komentar.

Apabila toolbar Edit belum ditampilkan, pilih menu **View > Toolbar > Edit** untuk menampilkan toolbar Edit.



Gambar 1.23 Menampilkan toolbar Edit.

1.6.2 Memenggal Kode Macro

Kode Macro dapat dipenggal ke baris berikutnya, jika Anda merasa kode Macro terlalu panjang. Anda juga dapat melakukan pemenggalan pada komentar. Pemenggalan dilakukan dengan spasi yang diikuti garis bawah (_). Perhatikan contoh kode Macro yang dipenggal berikut:

```
MsgBox "Nama worksheet sudah ada atau belum diisi",  
vbOKOnly + vbCritical, "Error Nama Worksheet"
```

Penulis lebih menyukai jika hasil pemenggalan Kode Macro ditempatkan lebih menjorok ke dalam menggunakan tombol **Indent** pada toolbar Edit. Kode Macro yang lebih menjorok ke dalam tidak berpengaruh saat Macro dijalankan.

```
MsgBox "Nama worksheet sudah ada atau belum diisi", _  
vbOKOnly + vbCritical, "Error Nama Worksheet"
```

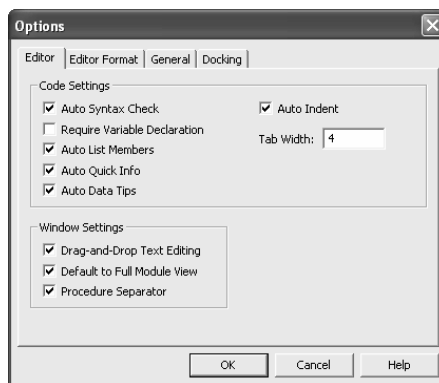
Selain pemenggalan kode Macro, penulis juga menuliskan kode Macro lebih menjorok ke dalam untuk memudahkan penulis saat menelusuri kode Macro yang jumlahnya cukup banyak, misalnya statement dalam struktur If Then berikut:

```
'Jika selLevelAkses berisi Admin  
If selLevelAkses.Text = "Admin" Then  
    'Menampilkan Form Akun Pengguna  
    formAkun.Show  
End If
```

1.6.3 Fitur Auto List Members

Saat menuliskan kode Macro, Anda dapat memanfaatkan fitur Auto List Members, yaitu fitur yang akan menampilkan daftar objek, koleksi objek, properti atau method yang dimiliki sebuah objek. Dengan menggunakan fitur Auto List Members, kesalahan penulisan objek, koleksi objek, properti ataupun method dapat diminimalkan. VBA secara default mengaktifkan fitur Auto List Members. Apabila tidak aktif, Anda dapat mengaktifkannya melalui kotak dialog Options. Berikut langkah-langkah pengaturan fitur Auto List Members:

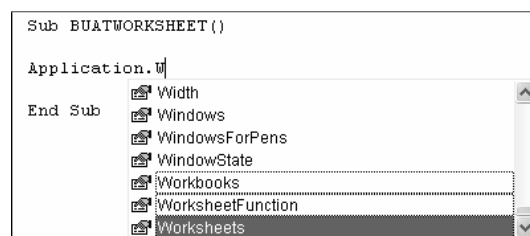
1. Pilih menu **Tools > Options...** kemudian muncul kotak dialog Options. Pilih tab **Editor**.



Gambar 1.24 Kotak dialog Options – tab Editor.

2. Beri tanda centang pada pilihan Auto List Members kemudian klik tombol **OK**.

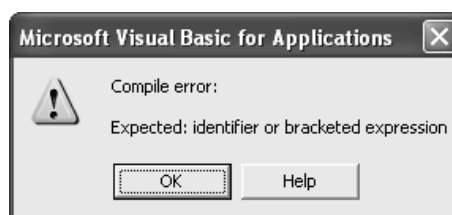
Untuk menampilkan fitur Auto List Members tekan kombinasi **Ctrl+J**. Daftar objek, koleksi objek, properti atau method yang dimiliki objek akan terlihat seperti pada Gambar 1.25. Anda dapat menekan tombol **Esc** pada keyboard untuk menyembunyikan fitur Auto List Members. Untuk memilih salah satu opsi yang ditampilkan daftar Auto List Members, tekan tombol **Tab** pada keyboard, atau klik ganda opsi yang ingin Anda pilih.



Gambar 1.25 Tampilan fitur Auto List Members.

1.6.4 Kesalahan Penulisan Kode Macro

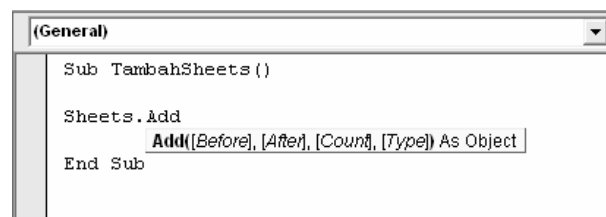
Apabila kode Macro yang Anda tulis salah, VBA secara default akan menampilkan kotak pesan peringatan. Klik tombol **OK** jika Anda mengetahui solusi untuk menangani kesalahan yang terjadi. Jika Anda memerlukan informasi bantuan untuk menangani kesalahan yang terjadi, klik tombol **Help**. VBA secara default mengaktifkan fitur ini. Jika tidak aktif, Anda dapat mengaktifkannya melalui kotak dialog Options tab Editor pilihan **Auto Syntax Check**.



Gambar 1.26 Kesalahan penulisan kode Macro.

1.6.5 Fitur Auto Quick Info

Fitur Auto Quick Info digunakan untuk menampilkan informasi argumen dari fungsi, properti atau method. VBA secara default mengaktifkan fitur Auto Quick Info. Apabila tidak aktif, Anda dapat mengaktifkannya melalui kotak dialog Options tab Editor pilihan **Auto Quick Info**.

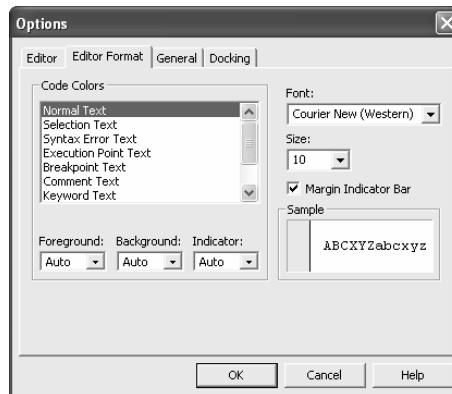


Gambar 1.27 Informasi yang ditampilkan Auto Quick Info.

1.6.6 Pengaturan Format Font Kode Macro

Secara subjektif, penulis merasa nyaman dengan pengaturan default format font kode Macro. Jika Anda kurang merasa nyaman dengan pengaturan default format font kode Macro, Anda dapat melakukan pengaturan format font kode Macro sendiri.

1. Pilih menu **Tools > Options...** kemudian muncul kotak dialog Options. Pilih tab **Editor Format**.
2. Berikut pilihan pengaturan format font kode Macro yang dapat Anda terapkan:
 - ❖ **Code Colors**. Pada pilihan ini, Anda dapat mengatur warna font kode Macro (**F**oreground), background teks kode Macro (**B**ackground) atau indikator kode Macro (**I**ndicator) elemen kode Macro pada daftar pilihan yang tersedia.
 - ❖ **Font**. Pada kotak pilihan **F**ont: Anda dapat mengatur jenis font elemen kode Macro pada daftar pilihan yang tersedia.
 - ❖ **Size**. Pada kotak pilihan **S**ize: Anda dapat mengatur ukuran font elemen kode Macro pada daftar pilihan yang tersedia.
 - ❖ **Margin Indicator Bar**. Beri tanda centang pada pilihan ini untuk menampilkan indikator visual saat *debugging* Macro.



Gambar 1.28 Kotak dialog Options – tab Editor Format.

1.7 OBJEK

Objek merupakan komponen dalam Macro yang mempunyai properti dan method sehingga dapat diubah atau dimanipulasi. Suatu objek dapat berupa satu objek atau kumpulan dari beberapa objek (koleksi objek). Sebuah koleksi juga merupakan objek sehingga mempunyai properti dan method yang dapat diubah (dimanipulasi). Objek dalam Macro mempunyai tingkatan dari yang paling umum sampai tingkatan yang paling khusus. Excel merupakan objek yang paling umum (Application) yang mempunyai tingkatan objek di bawahnya sebagai koleksi objek, misalnya Workbook, WorksheetFunction, dan koleksi objek lainnya.

Saat menuliskan kode Macro, objek dan koleksi objek dipisahkan menggunakan tanda titik (.). Untuk kondisi tertentu, Anda dapat menuliskan objek tanpa perlu menyebutkan objek tingkatan di atasnya, misalnya kode Macro untuk menyeleksi worksheet **Data** sebagai berikut:

```
Sheets("Data").Select
```

Apabila dalam waktu bersamaan terdapat dua buah workbook yang dibuka, di mana masing-masing workbook mempunyai worksheet Data, maka Anda harus menyebutkan objek tingkatan di atasnya (workbook) dalam kode Macro. Jika tidak disebutkan,

workbook yang dipilih adalah workbook yang sedang aktif. Berikut contoh kode Macro yang harus Anda tuliskan jika workbook yang Anda maksud adalah Analisis.xls:

```
Workbooks("Analisis.xls").Sheets("Data").Select
```

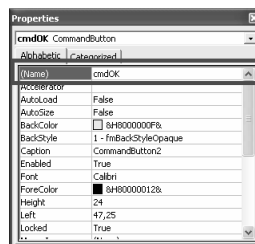
1.8 PROPERTI

Properti merupakan karakteristik yang dimiliki objek. Seperti halnya benda yang memiliki beberapa identitas, suatu objek juga memiliki beberapa properti yang menunjukkan identitas objek tersebut. Apabila suatu benda memiliki identitas panjang, lebar, tebal, warna dan identitas lainnya, maka objek VBA memiliki beberapa properti yang menunjukkan identitas, misalnya Name, Height, Width, BackColor atau Caption. Pengaturan properti objek sangat penting untuk membedakan satu objek dengan objek yang lain. Properti objek dapat diatur melalui window Properties atau melalui kode Macro saat *runtime*, yaitu saat Macro dijalankan.

1.8.1 Mengubah Properti melalui Window Properties

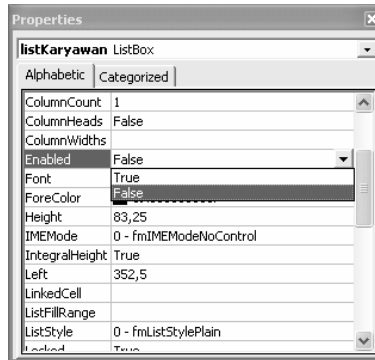
Untuk mengubah properti objek melalui window Properties, pastikan objek dalam keadaan terpilih. Hal ini perlu Anda perhatikan karena window Properties akan menampilkan daftar properti objek yang dipilih. Pengaturan properti melalui window Properties dilakukan dengan salah satu cara sebagai berikut:

- © Mengetikkan secara langsung nilai properti yang akan dimasukkan. Setelah Anda mengetikkan nilai properti, tekan tombol **Enter**. Contoh pengaturan properti yang dilakukan dengan cara mengetikkan secara langsung adalah Name, Height, Width, Left, Top.



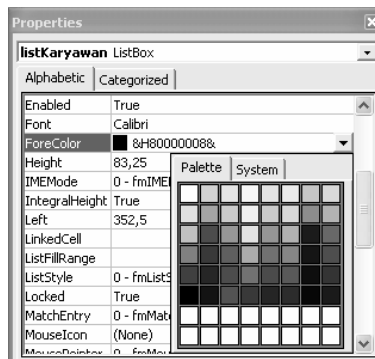
Gambar 1.29 Mengetikkan secara langsung properti Name.

- © Memilih salah satu opsi yang disediakan dalam bentuk kotak pilihan. Contoh properti yang ditentukan dengan cara seperti ini adalah Enabled atau Visible.




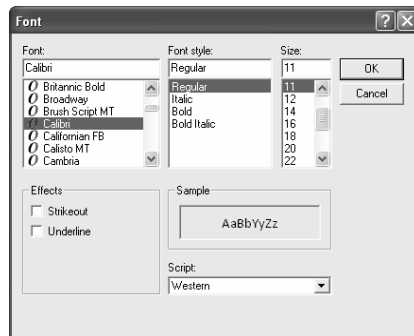
Gambar 1.30 Memilih opsi pengaturan properti Enabled.

- © Memilih salah satu opsi yang disediakan dalam bentuk daftar panel. Contoh properti yang ditentukan dengan cara seperti ini adalah BackColor, BorderColor, ForeColor.



Gambar 1.31 Memilih opsi pengaturan properti ForeColor.

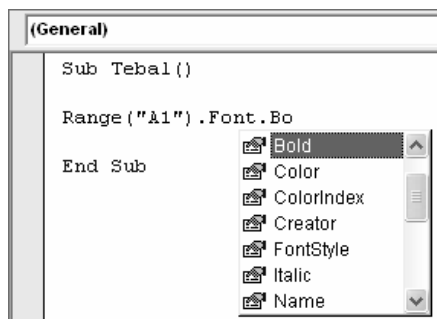
- © Menggunakan kotak dialog. Contoh properti yang pengaturannya ditentukan melalui kotak dialog adalah Font. Klik tombol  yang terletak pada properti Font kemudian muncul kotak dialog Font. Atur properti Font yang Anda inginkan. Jika sudah, klik tombol **OK**.



Gambar 1.32 Pengaturan properti Font melalui kotak dialog.

1.8.2 Mengubah Properti Saat Runtime

Properti objek juga dapat diubah saat *runtime* menggunakan kode Macro. Saat menuliskan kode Macro, objek dan properti dipisahkan menggunakan tanda titik (.). Anda dapat memanfaatkan fitur Auto List Members untuk menampilkan daftar properti sebuah objek.



Gambar 1.33 Daftar properti yang ditampilkan.

1.8.3 Properti yang Sering Digunakan

Beberapa properti dimiliki oleh seluruh objek, namun beberapa properti yang lain hanya dimiliki objek tertentu secara khusus. Berikut daftar properti objek penting yang sering digunakan:

- © **Name** merupakan properti untuk menentukan nama objek. Properti Name dimiliki semua objek. Nama objek yang Anda ubah harus sesuai dengan aturan dalam Visual Basic, misalnya nama objek tidak boleh menggunakan karakter < atau >. Selain adanya aturan tersebut, nama objek juga harus bersifat unik (tidak ada nama objek yang sama dalam sebuah workbook).
- © **BackStyle** merupakan properti yang digunakan untuk menentukan style objek, apakah transparan atau tidak. Pilihan yang disediakan adalah:

Konstanta	Nilai	Keterangan
fmBackStyleTransparent	0	Objek terlihat transparan. Pada pilihan ini pengaturan properti BackColor tidak berlaku.
fmBackStyleOpaque	1	Objek terlihat solid (tidak transparan). Pada pilihan ini pengaturan properti BackColor berlaku.

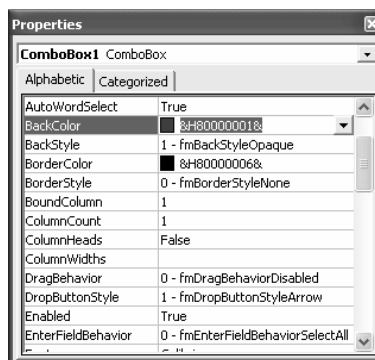
- © **BackColor** merupakan properti yang digunakan untuk menentukan warna latar belakang (background) objek. Warna latar belakang objek ditampilkan jika properti BackStyle diatur ke **fmBackStyleOpaque**. Untuk mengubah properti BackColor melalui kode Macro, Anda dapat memilih warna konstan yang disediakan Visual Basic:

Konstanta	Nilai	Keterangan
vbBlack	0x0	Warna hitam
vbRed	0xFF	Warna merah
vbGreen	0xFF00	Warna hijau
vbYellow	0xFFFF	Warna kuning
vbBlue	0xFF0000	Warna biru
vbMagenta	0xFF00FF	Warna magenta
vbCyan	0xFFFF00	Warna biru muda
vbWhite	0FFFFFFF	Warna putih

Anda juga dapat mengubah warna latar belakang objek menggunakan kombinasi warna merah, hijau dan biru (RGB). Nilai masing-masing warna berkisar antara 0 sampai 255. Berikut contoh kode Macro untuk mengubah warna latar belakang objek menggunakan RGB:

```
Sheets("Sheet1").chkPria.BackColor = RGB(100, 0, 10)
```

Cara termudah untuk mengubah properti BackColor menggunakan kode Macro adalah dengan teknik *copy paste* properti BackColor pada window Properties. Teknik ini juga dapat Anda terapkan untuk properti lain yang ada hubungannya dengan pengaturan warna. Blok properti BackColor pada window Properties. Tekan kombinasi tombol **Ctrl+C** untuk mengcopy properti BackColor.



Gambar 1.34 Copy properti BackColor.


Ketikkan kode Macro `Sheets("Sheet1").chkPria.BackColor =` kemudian tekan kombinasi tombol **Ctrl+V**. Kode Macro hasil *copy paste* dari window Properties dapat dilihat pada contoh berikut ini:

```
Sheets("Sheet1").chkPria.BackColor = &H80000001
```

- © **Caption** merupakan properti untuk menentukan teks yang ditampilkan objek. Pengaturan properti Caption (teks yang ditampilkan) tidak harus sama dengan pengaturan properti Name (nama objek).

- © **Enabled** merupakan properti untuk menentukan apakah objek akan diaktifkan atau tidak. Apabila objek tidak aktif, objek tidak dapat merespons event seperti klik atau klik ganda. Pilihan yang disediakan adalah:

Konstanta	Keterangan
True	Objek aktif dan dapat merespons event.
False	Objek tidak aktif dan tidak dapat merespon event. Objek yang tidak aktif ditandai tampilan teks (nilai) atau gambar yang menjadi buram.

- © **Font** merupakan properti untuk menentukan jenis, style, ukuran, dan efek font pada teks (nilai) dalam objek. Untuk mengubah properti Font melalui window Properties klik tombol  pada properti **Font** kemudian muncul kotak dialog Font. Pilih jenis, style, ukuran, dan efek font yang Anda inginkan. Jika sudah, klik tombol **OK**.

Untuk mengubah properti Font menggunakan kode Macro, masing-masing subproperti Font memiliki pengaturan sendiri.

Subproperti	Keterangan
Name	Untuk menentukan jenis font. Pengaturan dilakukan dengan menuliskan jenis font yang akan digunakan dengan diapit tanda petik.
Bold	Untuk mengatur apakah Font menampilkan huruf tebal atau normal. Pilihannya adalah TRUE (huruf tebal) atau FALSE (huruf normal).
Italic	Untuk mengatur apakah Font menampilkan huruf miring atau normal. Pilihannya adalah TRUE (huruf miring) atau FALSE (huruf normal).
Underline	Untuk mengatur apakah Font menampilkan huruf dengan garis bawah. Pilihannya adalah TRUE (huruf dengan garis bawah) atau FALSE (huruf tanpa garis bawah).
Size	Untuk menentukan ukuran font. Pengaturan dilakukan dengan menuliskan ukuran font yang akan digunakan.
Strikethrough	Untuk mengatur apakah Font menampilkan huruf dengan tanda coret. Pilihannya adalah TRUE (huruf dengan tanda coret) atau FALSE (huruf tanpa tanda coret).


Weight	Untuk mengatur ketebalan font. Pengaturan dilakukan dengan menuliskan ketebalan font yang akan digunakan dari 1 sampai 1000.
--------	--

Berikut contoh kode Macro untuk melakukan pengaturan masing-masing subproperti font:

```

Sheets("Sheet1").chkPria.Font.Name = "Times new Roman"
Sheets("Sheet1").chkPria.Font.Bold = False
Sheets("Sheet1").chkPria.Font.Italic = True
Sheets("Sheet1").chkPria.Font.Underline = True
Sheets("Sheet1").chkPria.Font.Size = 12
Sheets("Sheet1").chkPria.Font.Strikethrough = True
Sheets("Sheet1").chkPria.Font.Weight = 50

```

- ◎ **ForeColor** merupakan properti untuk menentukan warna font (teks/nilai) pada objek. Untuk mengubah properti ForeColor melalui window Properties, klik *drop down* properti ForeColor kemudian pilih warna yang Anda inginkan. Untuk mengubah properti ForeColor melalui kode Macro, Anda dapat memilih warna yang disediakan Visual Basic. Anda juga dapat mengubah warna properti ForeColor menggunakan kombinasi warna merah, hijau dan biru (RGB).
- ◎ **Height** merupakan properti untuk mengatur tinggi objek.
- ◎ **Left** merupakan properti untuk mengatur jarak tepi bidang kontrol di sebelah kiri dengan tepi bidang sebelah kiri tempat penampung kontrol (worksheet atau UserForm).
- ◎ **Picture** merupakan properti untuk menentukan gambar yang akan ditampilkan pada objek. Untuk menentukan properti Picture melalui window Properties, klik tombol  pada properti **Picture** kemudian muncul kotak dialog Load Picture. Pilih file foto (gambar) yang akan digunakan. Jika sudah, klik **OK**.



Gambar 1.35 Kotak dialog Load Picture.

Selain melalui window Properties, Anda juga dapat menentukan properti Picture menggunakan kode Macro. Pengaturan properti dilakukan menggunakan fungsi `LoadPicture` dengan argumen alamat penyimpanan file gambar yang dituliskan secara lengkap. Berikut contoh kode Macro untuk menentukan properti Picture:

```

=
Sheets("Sheet1").Image1.Picture
LoadPicture("E:\NX_G911.jpg")

```

- ⊙ **Top** merupakan properti untuk mengatur jarak tepi atas bidang kontrol dengan tepi bidang atas tempat penampung kontrol (worksheet atau UserForm).
- ⊙ **Visible** merupakan properti untuk menentukan apakah objek akan ditampilkan atau tidak. Pilihan yang disediakan adalah:

Konstanta	Keterangan
True	Objek ditampilkan.
False	Objek tidak ditampilkan.

- ⊙ **Width** merupakan properti untuk mengatur lebar objek.

1.9 METHOD

Method merupakan suatu set perintah seperti halnya Function Procedure dan Sub Procedure, tetapi sudah tersedia di dalam suatu objek. Penggunaan method dalam kode Macro akan tergantung pada kaitan perintah dan jumlah argumen yang diperlukan serta apakah method tersebut mengembalikan suatu nilai.

1.10 OPERATOR VBA

Operator merupakan simbol yang digunakan untuk melakukan suatu operasi nilai data. Simbol operator bisa berupa karakter ataupun kata khusus.

1.10.1 Operator Aritmetika

Operator aritmetika digunakan untuk melakukan kalkulasi operasi matematika seperti penambahan, pengurangan, perkalian, pembagian, perpangkatan, dan persentase.

Operator aritmetika	Simbol	Contoh	Hasil
Penambahan	+	10+5	15
Pengurangan	-	10-5	5
Perkalian	*	10*5	50
Pembagian	/	10/5	2
Perpangkatan	^	10^5	100.000
Persentase	%	10%	0,10

Apabila dalam suatu proses perhitungan terdapat beberapa operator aritmetika, urutan prioritas perhitungan sebagai berikut:

- ⊙ Perhitungan yang diapit tanda kurung dilakukan paling dulu.
- ⊙ Pada level yang sama, urutan perhitungan dilakukan menurut jenis operator. Perpangkatan dilakukan lebih dulu baru perkalian, pembagian, penambahan dan pengurangan.
- ⊙ Perhitungan pada level yang sama dilakukan dari sebelah kiri dulu.

1.10.2 Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan dua nilai. Ketika dua nilai tersebut dibandingkan menggunakan operator perbandingan, maka akan menghasilkan nilai logika benar (*true*) atau salah (*false*).

Operator perbandingan	Simbol	Contoh	Hasil
Sama dengan	=	10=5	False
Lebih besar	>	10>5	True
Lebih kecil	<	10<5	False
Lebih besar sama dengan	>=	10>=5	True

Lebih kecil sama dengan	<=	10<=5	False
Tidak sama dengan	<>	10<>5	True

1.10.3 Operator Teks

Operator teks digunakan untuk menghubungkan atau menggabungkan dua nilai teks sehingga menghasilkan satu gabungan nilai teks.

Operator	Simbol	Contoh	Hasil
Menggabungkan dua nilai teks sehingga menghasilkan satu gabungan nilai teks.	&	"Aku " & "malu"	Aku malu

1.10.4 Operator Referensi

Operator referensi digunakan untuk menggabungkan sel (range) atau sebagai pemisah argumen.

Operator referensi	Simbol	Contoh
Operator range	:	B1:B5
Operator pemisah argumen	; atau ,	=COUNTIF(B1:B5;"Pria")

1.10.5 Operator Logika

Operator logika digunakan untuk mengekspresikan satu atau lebih data logika yang menghasilkan data logika baru. Operator logika juga dapat digunakan untuk melakukan suatu operasi pengambilan keputusan pada program yang dijalankan. Operator logika akan menghasilkan nilai benar (*true*) atau salah (*false*).

Operator logika	Keterangan	Contoh
And	Dan	A1 And A2
Not	Tidak	Not A2
Or	Atau	A1 Or A2

1.11 VARIABEL

Variabel merupakan tempat dalam memori komputer yang diberi nama sebagai pengenalan dan dialokasikan untuk menampung data. Sesuai data yang ditampung, variabel harus mempunyai tipe data yang sesuai dengan isinya. Secara default, tipe data yang digunakan dalam variabel adalah variant. Jika Anda tidak mengetahui dengan pasti tipe data dalam variabel yang akan digunakan, tipe data sebaiknya dikosongkan. Deklarasi variabel harus diletakkan sebelum baris perintah yang menggunakan variabel tersebut.

Dalam mendeklarasikan variabel, Anda perlu memerhatikan jangkauan dari variabel tersebut. Jangkauan variabel pada VBA dapat diketahui dengan kata kunci **Public**, **Private** dan **Dim** pada saat variabel dideklarasikan. Variabel yang dideklarasikan dengan kata kunci **Public** akan tersedia bagi semua Procedure di semua Module dalam suatu proyek, tempat di mana variabel tersebut dideklarasikan. Variabel yang dideklarasikan dengan kata kunci **Private** akan tersedia bagi semua Procedure dalam Module tempat di mana variabel tersebut dideklarasikan. Variabel yang dideklarasikan dengan kata kunci **Dim** hanya akan tersedia dalam Procedure, tempat di mana variabel tersebut dideklarasikan. Format kode Macro dalam mendeklarasikan variabel adalah sebagai berikut:

```
Jangkauan NamaVariabel As TipeData
```

Berikut contoh pendeklarasian sebuah variabel:

```
Public Umur As Single  
Private Nama As String  
Dim DataProduksi As Range
```

1.12 KONSTANTA

Konstanta adalah nama yang menyimpan suatu nilai yang tidak dapat berubah. Kecepatan proses pada konstanta lebih cepat dibandingkan variabel karena tidak perlu menunggu pengisian data. Dalam mendeklarasikan konstanta, Anda perlu memerhatikan jangkauan dari konstanta tersebut. Jangkauan konstanta pada VBA dapat diketahui dengan kata kunci **Public**, **Private** dan **Const** pada saat konstanta dideklarasikan. Konstanta yang dideklarasikan dengan kata kunci **Public** akan tersedia bagi semua Procedure di semua Module dalam suatu proyek, tempat di mana

konstanta tersebut dideklarasikan. Konstanta yang dideklarasikan dengan kata kunci **Private** akan tersedia bagi semua Procedure dalam Module tempat di mana konstanta tersebut dideklarasikan. Konstanta yang dideklarasikan dengan kata kunci **Const** hanya akan tersedia dalam Procedure tempat di mana konstanta tersebut dideklarasikan. Format kode Macro untuk mendeklarasikan konstanta adalah sebagai berikut:

```
Jangkauan NamaKonsanta As TipeData = Ekspresi
```

Berikut contoh pendeklarasian sebuah konstanta:

```
Public Const Berat As Integer = 50  
Private Const Nama As String = "Yudhy Wicaksono"  
Const Penjualan As Integer = 25000
```

1.13 ARRAY

Array merupakan variabel yang mampu menyimpan beberapa nilai dengan tipe data yang sama. Kumpulan nilai tersebut satu sama lain dibedakan dengan indeks dan masing-masing disebut elemen array. Beberapa nilai data yang mempunyai tipe data sama akan lebih mudah jika dimasukkan ke dalam sebuah array dibandingkan dimasukkan dalam beberapa variabel yang berbeda. Ada dua jenis array, yaitu *fixed array* dan *dynamic array*. *Fixed array* merupakan array yang ukurannya tetap. Format kode Macro dalam mendeklarasikan *fixed array* adalah sebagai berikut:

```
Jangkauan NamaArray (Indeks) As TipeData
```

Indeks merupakan jumlah elemen yang akan digunakan pada array. Anda dapat menuliskan sebuah *fixed array* dengan cara berikut ini:

```
Dim harga As Single  
NamaBuah = Array(2500, 3500, 5000)
```

Anda juga dapat menuliskan *fixed array* dengan cara berikut ini:

```
Dim harga(2) As Single  
NamaBuah(0) = 2500  
NamaBuah(1) = 3500  
NamaBuah(2) = 5000
```

Sebuah array dibatasi oleh batas bawah dan batas atas. Secara default, batas bawah array adalah nol (0). Apabila batas bawah array adalah nol (0), maka pada Indeks Anda isikan dengan jumlah elemen dikurangi satu. Misalnya Anda akan menggunakan 3 elemen array, maka pada saat mendeklarasikan array, **Indeks** Anda isikan dengan angka 2. Selain menggunakan batas bawah nol (0), Anda juga dapat mengubah batas bawah array dengan angka satu (1). Berikut cara yang dapat Anda lakukan untuk mendeklarasikan batas bawah array:

1. Menggunakan pernyataan **Option Base**. Format kode Macro dalam menggunakan **Option Base** adalah sebagai berikut:

```
Option Base 1
```

Misalnya, Anda ingin menentukan batas bawah array adalah 1, maka sebelum pendeklarasian array Anda tuliskan kode Macro:

```
Option Base 1  
Dim namabuah(3) As String
```

Anda tidak perlu menuliskan pernyataan Option Base, apabila Anda menggunakan nol (0) sebagai batas bawah array.

2. Menggunakan pernyataan **To**. Format kode Macro dalam menggunakan pernyataan **To** adalah sebagai berikut:

```
KataKunci NamaArray(BatasBawah To BatasAtas) As TipeData
```

Misalnya, Anda ingin menentukan batas bawah array adalah 1 dan batas atas array adalah 3, maka kode Macro yang Anda tuliskan untuk mendeklarasikan array adalah sebagai berikut:

```
Dim namabuah(1 To 3) As String
```

Berbeda dengan *fixed array*, ukuran dalam *dynamic array* dapat berubah. *Dynamic array* sangat berguna dalam pemrograman yang jumlah elemen *array*nya tidak bisa diketahui sejak awal. Format kode Macro dalam mendeklarasikan *dynamic array* adalah sebagai berikut:

```
KataKunci NamaArray() As TipeData
```


Berikut contoh penulisan sebuah *dynamic array*:

```
Dim harga() As Single
```

1.14 TIPE DATA

Setiap variabel, konstanta atau array yang dideklarasikan dalam VBA mempunyai tipe data. Pemilihan tipe data akan menentukan nilai apa yang dapat ditampung oleh variabel, konstanta atau array. Oleh karena itu, pemilihan tipe data harus tepat dan sesuai dengan nilai yang akan ditampung. Secara default, tipe data yang digunakan dalam variabel, konstanta atau array adalah variant. Jika Anda tidak mengetahui dengan pasti tipe data yang akan digunakan, maka tipe data sebaiknya dikosongkan atau diisi dengan tipe data variant. Berikut beberapa tipe data dalam VBA:

Tipe data	Ukuran	Contoh
Integer	2 byte	Semua bilangan antara -32.768 sampai 32.767
Long	4 byte	Semua bilangan antara -2.147.483.648 sampai 2.147.483.648
Single	4 byte	Bilangan negatif antara $-3,402823 \times 10^{38}$ sampai $-1,401298 \times 10^{-45}$ Bilangan positif antara $1,401298 \times 10^{-45}$ sampai $3,402823 \times 10^{38}$
Double	8 byte	Bilangan negatif antara $-1,7976931348623 \times 10^{308}$ sampai $-4,940656458623 \times 10^{-324}$ Bilangan positif antara $4,940656458623 \times 10^{-324}$ sampai $1,7976931348623 \times 10^{308}$
Decimal	14 byte	$\pm 79228162514264337593543950335$ (tanpa titik desimal) $\pm 7,9228162514264337593543950335$ (dengan 28 angka di belakang titik desimal)
Currency	8 byte	Bilangan dengan nilai antara -922.337.203.685.477,5808 sampai 922.337.203.685.477,5808
String	1 byte	Untuk menyimpan teks berisi 0 sampai 2 miliar karakter.

Byte	1 byte	Bilangan antara 0 sampai 255.
Boolean	2 byte	Berisi nilai True (benar) atau False (salah).
Date	8 byte	Menyimpan informasi tanggal dan waktu. Tanggal antara 1 Januari 100 sampai 31 Desember 9999. Waktu antara 00:00:00 sampai 23:59:59.
Object	4 byte	Digunakan untuk mengakses objek apa saja yang diperlukan oleh VBA, disimpan dalam alamat memory objek tersebut.
Variant	16 byte	Seluruh tipe data yang ada. Jika berupa teks, maka akan disimpan dalam bentuk teks. Jika berupa bilangan, maka akan disimpan dalam tipe Double.

1.15 BEKERJA DENGAN NAMA RANGE

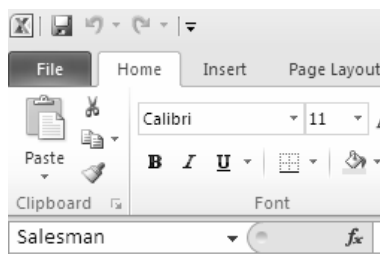
Nama range merupakan sebuah nama yang didefinisikan (diberikan) pada suatu range yang kita pilih. Penggunaan nama range akan sangat membantu Anda ketika membuat aplikasi menggunakan Macro Excel. Anda tidak perlu mengubah kode Macro jika range yang digunakan dalam aplikasi dipindahkan. Hal tersebut tidak akan berlaku apabila Anda menggunakan alamat referensi sel atau range. Nama range dapat dibuat dengan tiga cara, yaitu melalui Name Box pada Formula Bar, kotak dialog Create Names from Selection dan kotak dialog New Name. Sebelum membuat nama range, Anda sebaiknya mengetahui beberapa aturan pembuatan nama range sebagai berikut:

- ☉ Karakter yang boleh digunakan dalam nama range adalah tanda titik (.) dan *underscore* (_).
- ☉ Nama range harus dimulai dari huruf atau tanda *underscore* (_), bukan angka atau karakter lainnya.
- ☉ Nama range yang terdiri atas dua kata atau lebih tidak boleh dipisahkan dengan spasi. Anda dapat menggunakan tanda *underscore* (_) atau titik (.) untuk menggantikan spasi. Misalnya, **Nama barang** dapat Anda ganti dengan **Nama_barang** atau **Nama.barang**.
- ☉ Dalam satu workbook, nama range harus bersifat unik (tidak ada nama range yang sama dalam satu workbook).

1.15.1 Membuat Nama Range melalui Name Box

Cara termudah untuk membuat nama range adalah melalui Name Box pada Formula Bar. Walaupun relatif mudah, nama range yang dibuat melalui Name Box mempunyai beberapa keterbatasan, misalnya kita tidak dapat membuat nama range yang bersifat dinamis.

1. Buka workbook **Membuat Nama Range.xlsx** yang disertakan dalam Bonus CD buku ini.



Gambar 1.36 Mengetikkan nama range dalam Name Box.

2. Untuk membuat nama range A3:A63 yang berisi data nama salesman, blok range **A3:A63**. Arahkan cursor mouse pada Name Box kemudian ketikkan **Salesman**. Tekan tombol **Enter** pada keyboard.

A screenshot of an Excel spreadsheet titled 'Database Penjualan Peralatan Kantor'. The table has columns for Salesman, Tanggal, Nama barang, Jumlah, Harga, and Total. The data is organized by salesman, with each salesman's entries grouped together. The salesmen listed are Firman, Agus, Wulan, Dinu, Rahmat, and Citra. The table contains 20 rows of data, with the last row (row 20) showing data for Citra on 10/01/2010 for a 'Pita Mesin Tik Elektrik' with a quantity of 5, a price of 25.000, and a total of 125.000.

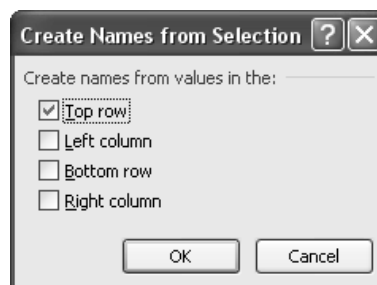
Salesman	Tanggal	Nama barang	Jumlah	Harga	Total
Firman	01/01/2010	Kertas HVS F4	5	30.000	150.000
Firman	01/01/2010	Trigonal Clip	10	16.000	160.000
Agus	01/01/2010	Clear Holder Isi 20 Lbr	7	8.000	56.000
Wulan	01/01/2010	Ordner Kertas Folio	5	72.500	362.500
Wulan	01/01/2010	Kertas HVS A4	8	28.500	228.000
Firman	03/01/2010	Pita Mesin Tik	1	8.000	8.000
Dinu	03/01/2010	Spidol Kecil	10	8.500	85.000
Dinu	04/01/2010	Pita Mesin Tik Elektrik	2	25.000	50.000
Rahmat	05/01/2010	Stapler HD 10	3	6.500	19.500
Wulan	05/01/2010	Buku Block Note	6	2.500	15.000
Firman	05/01/2010	Spidol Kecil	3	8.500	25.500
Firman	05/01/2010	Trigonal Clip	10	16.000	160.000
Citra	06/01/2010	Kertas Buram F4	10	9.500	95.000
Citra	07/01/2010	Pita Mesin Tik	5	8.000	40.000
Citra	08/01/2010	Pita Mesin Absensi	5	75.000	375.000
Citra	08/01/2010	Kertas HVS A4	2	28.500	57.000
Agus	08/01/2010	Clear Holder Isi 20 Lbr	10	8.000	80.000
Citra	10/01/2010	Pita Mesin Tik Elektrik	5	25.000	125.000

Gambar 1.37 Membuat nama range melalui Name Box

1.15.2 Membuat Nama Range melalui Kotak Dialog Create Names from Selection

Kotak dialog Create Names from Selection digunakan untuk membuat nama pada range yang diseleksi. Nama range yang digunakan adalah data dalam sel pada baris teratas (*top row*), baris terbawah (*bottom row*), kolom paling kiri (*left column*) atau kolom paling kanan (*right column*) dari range yang diseleksi. Jika data dalam sel yang akan digunakan sebagai nama range mempunyai spasi, secara otomatis spasi tersebut akan diganti dengan tanda *underscore* (_). Dalam contoh kali ini kita akan membuat nama range Tanggal.

1. Blok range **B2:B63** yang berisi data tanggal. Klik tombol **Create from Selection** dalam tab Formulas group Defined Names. Muncul kotak dialog Create Names from Selection. Kotak dialog Create Names from Selection juga dapat ditampilkan dengan menekan kombinasi tombol **Ctrl+Shift+F3** secara bersamaan.
2. Beri tanda centang pada pilihan **Top row** untuk menggunakan data dalam sel baris paling atas range yang diseleksi sebagai nama range, yaitu Tanggal. Klik tombol **OK**.



Gambar 1.38 Kotak dialog Create Names from Selection.

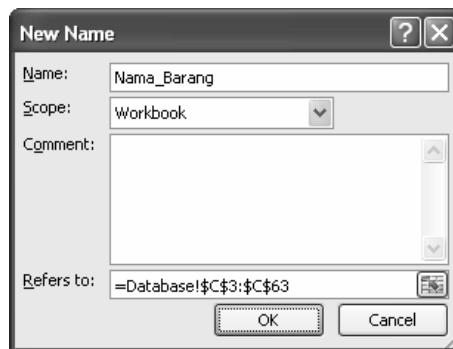
1.15.3 Membuat Nama Range melalui Kotak Dialog New Name

Selain melalui Name Box dan kotak dialog Create Names from Selection, Anda juga dapat membuat nama range melalui kotak

dialog New Name. Salah satu kelebihan kotak dialog New Name adalah kita dapat mengatur cakupan (*scope*) nama range yang akan dibuat. Kelebihan lainnya adalah adanya pengaturan nama range yang lebih luas, misalnya kita dapat mengatur apakah nama range yang dibuat bersifat statis atau dinamis.


Apabila Anda memilih Workbook, maka nama range dapat digunakan untuk seluruh worksheet. Jika Anda memilih salah satu worksheet untuk membuat nama range, maka untuk menggunakan nama range pada worksheet yang berbeda, Anda harus menuliskan nama worksheet diikuti tanda seru (!). Misalnya, Anda membuat nama range **Penjualan** pada worksheet **Sheet1**. Untuk menghitung jumlah pada worksheet selain Sheet1, maka formula yang harus Anda ketikkan adalah **=Sum(Sheet1!Penjualan)**.

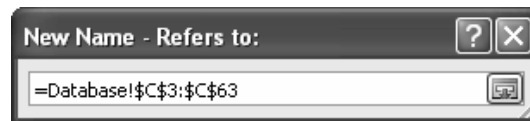
1. Tekan kombinasi tombol **Ctrl+F3** atau klik tombol **Name Manager** dalam tab Formulas group Defined Names. Muncul kotak dialog Name Manager.
2. Untuk membuat nama range baru, klik tombol **New...** Muncul kotak dialog New Name. Anda juga dapat menampilkan kotak dialog New Name secara langsung dengan cara klik tombol **Define Name** dalam tab Formulas group Defined Names.




Gambar 1.39 Kotak dialog New Name.

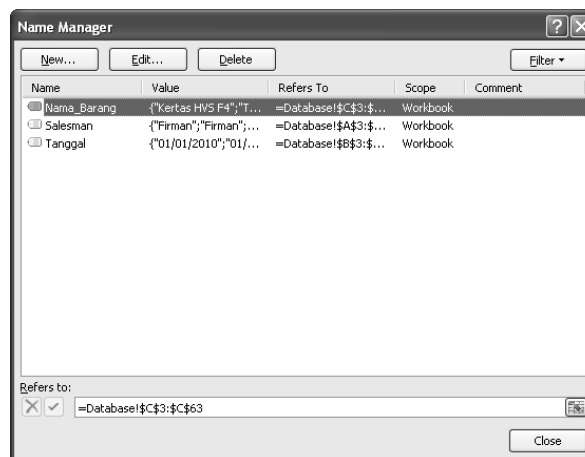
3. Ketikkan nama range yang akan dibuat pada kotak isian **Name:**, misalnya ketikkan **Nama_Barang**.
4. Pada kotak isian Scope: pilih **Workbook**, yang berarti nama range Nama_barang dapat digunakan untuk seluruh worksheet.

5. Ketikkan **=Database!\$C\$3:\$C\$63** pada kotak isian Refers to: Dalam contoh tersebut, range yang dibuat adalah range C3:C63 pada worksheet Database. Langkah tersebut juga dapat dilakukan dengan cara klik tombol  pada kotak sebelah kanan Refers to:.. Muncul *collapse dialog* seperti terlihat pada Gambar 1.40.



Gambar 1.40 Collapse dialog New Name – Refers to:..

6. Blok range **C3:C63** (hasil blok ditandai dengan garis putus-putus). Klik tombol . Muncul kembali kotak dialog New Name kemudian klik tombol **OK**.
7. Nama range yang sudah dibuat akan dimasukkan dalam daftar pada kotak dialog Nama Manager seperti pada Gambar 1.41. Untuk membuat nama range yang lain, lakukan dengan cara yang sama seperti pada langkah 2 sampai 6.

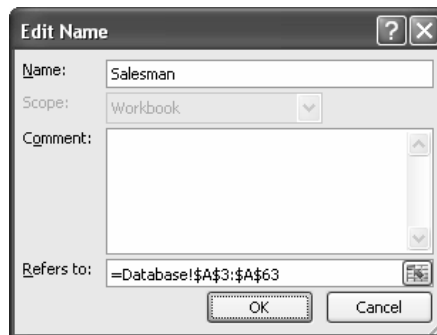


Gambar 1.41 Kotak dialog Name Manager.

1.15.4 Edit Nama Range

Nama range yang sudah Anda buat tidaklah bersifat baku. Anda dapat mengubah nama range sesuai kebutuhan. Edit nama range dilakukan melalui kotak dialog Nama Manager.

1. Klik tombol **Name Manager** dalam tab Formulas group Defined Names. Muncul kotak dialog Name Manager.
2. Pilih nama sel atau nama range yang akan diedit kemudian klik tombol **Edit...** Muncul kotak dialog Edit Name.



Gambar 1.42 Kotak dialog Edit Name.

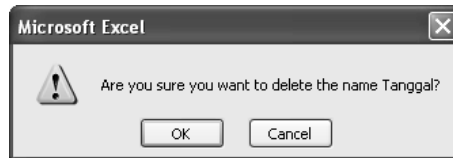
3. Cara mengedit nama range pada prinsipnya sama dengan cara membuat nama range. Perbedaannya ialah Anda tidak dapat mengedit Scope (cakupan) nama range. Lakukan pengeditan yang diperlukan kemudian klik tombol **OK**.

1.15.5 Menghapus Nama Range

Apabila Anda sudah tidak membutuhkan nama sel atau nama range, Anda dapat menghapusnya melalui kotak dialog Name Manager. Sebelum menghapus nama range, pastikan nama range yang akan dihapus memang sudah benar-benar tidak digunakan. Jika dalam workbook masih terdapat formula atau fungsi yang menggunakan nama range, baik secara langsung ataupun tidak langsung, maka perhitungan akan menghasilkan error **#REF!**.

1. Klik tombol **Name Manager** dalam tab Formulas group Defined Names. Muncul kotak dialog Name Manager.

2. Pilih nama sel atau nama range yang akan dihapus kemudian klik tombol **Delete...** Muncul kotak pesan seperti Gambar 1.43.



Gambar 1.43 Kotak pesan informasi.

3. Untuk menghapus nama range, klik tombol **OK**. Klik tombol **Cancel** untuk membatalkan.

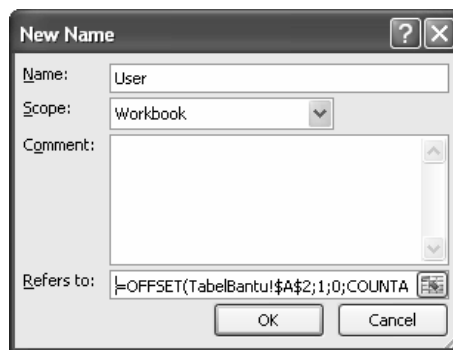
1.15.6 Membuat Nama Range Dinamis

Anda juga dapat membuat nama untuk range yang bersifat dinamis. Kelebihan nama range dinamis adalah range yang dipilih akan selalu menyesuaikan dengan data yang baru dimasukkan atau dihapus. Dalam contoh berikut, kita akan membuat nama range dinamis **User** dan **Tabel**.

1. Buka file **Membuat Nama Range Dinamis.xlsx** yang disertakan dalam Bonus CD buku ini.
2. Untuk membuat nama range **User**, klik tombol **Define Name** dalam tab **Formulas** group **Defined Names**. Muncul kotak dialog **New Name**.
3. Ketikkan **User** pada kotak isian **Name**:. Pilih **Workbook** pada kotak isian **Scope**: Pada kotak isian **Refers to**: ketikkan formula **=OFFSET(TabelBantu!\$A\$2;1;0;COUNTA(TabelBantu!\$A:\$A)-2;1)**. Klik tombol **OK**. Penjelasan formula pada kotak isian **Refers to**: adalah sebagai berikut:
 - **OFFSET** merupakan fungsi yang akan menyalin hasil yang ada dalam suatu range dengan jumlah baris dan kolom yang disebutkan jaraknya dari sel acuan.
 - **TabelBantu!\$A\$2** merupakan sel acuan (sel A2) yang dijadikan dasar dalam penggunaan fungsi **OFFSET**.
 - **1** menunjukkan arah baris ke bawah (positif). Karena nilainya 1, maka sel yang ditunjuk adalah sel 1 baris di

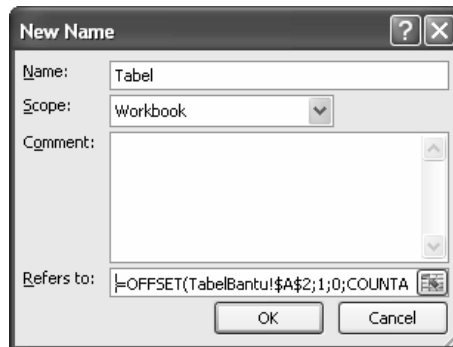
bawah sel yang dijadikan acuan dalam penggunaan fungsi OFFSET.

- **0** merupakan arah kolom. Karena nilainya 0, maka sel yang ditunjuk terletak dalam kolom yang sama dengan sel yang dijadikan acuan dalam penggunaan fungsi OFFSET.
- **COUNTA(TabelBantu!\$A:\$A)-2** merupakan tinggi range yang diwakili jumlah baris pada range dalam kolom A yang tidak kosong. Karena ada sel yang tidak digunakan dalam range (sel A1 dan A2) maka tinggi range dikurangi 2.
- **1** merupakan lebar range yang diwakili jumlah kolom pada range.



Gambar 1.44 Membuat nama range User.

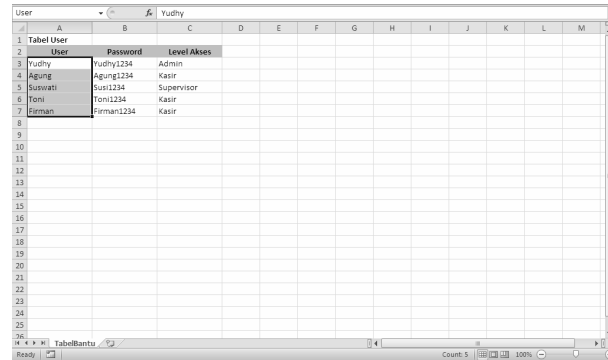
4. Untuk membuat nama range Tabel, klik tombol **Define Name** dalam tab Formulas group Defined Names. Muncul kotak dialog New Name.
5. Ketikkan **Tabel** pada kotak isian Name:. Pilih **Workbook** pada kotak isian Scope: Pada kotak isian Refers to: ketikkan formula **=OFFSET(TabelBantu!\$A\$2;1;0;COUNTA(TabelBantu!\$A:\$A)-2;3)**. Klik tombol **OK**. Penjelasan formula pada kotak isian Refers to: adalah sebagai berikut.



Gambar 1.45 Membuat nama range Tabel.

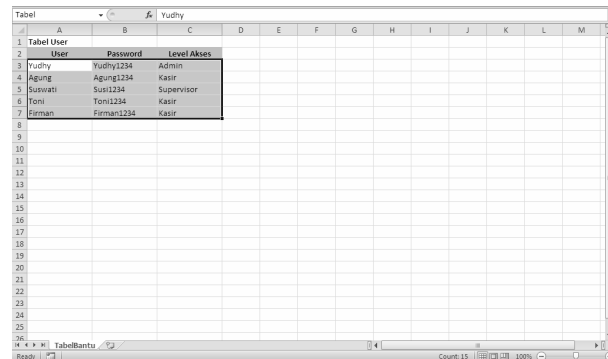
- **OFFSET** merupakan fungsi yang akan menyalin hasil yang ada dalam suatu range dengan jumlah baris dan kolom yang disebutkan jaraknya dari sel acuan.
 - **TabelBantu!\$A\$2** merupakan sel acuan (sel A2) yang dijadikan dasar dalam penggunaan fungsi **OFFSET**.
 - **1** menunjukkan arah baris ke bawah (positif). Karena nilainya 1, maka sel yang ditunjuk adalah sel 1 baris di bawah sel yang dijadikan acuan dalam penggunaan fungsi **OFFSET**.
 - **0** merupakan arah kolom. Karena nilainya 0, maka sel yang ditunjuk terletak dalam kolom yang sama dengan sel yang dijadikan acuan dalam penggunaan fungsi **OFFSET**.
 - **COUNTA(TabelBantu!\$A:\$A)-2** merupakan tinggi range yang diwakili jumlah baris pada range dalam kolom A yang tidak kosong. Karena ada sel yang tidak digunakan dalam range (sel A1 dan A2) maka tinggi range dikurangi 2.
 - **3** merupakan lebar range yang diwakili jumlah kolom pada range. Klik tombol **OK**.
6. Untuk menguji hasilnya, masukkan record data baru ke dalam tabel dengan langkah-langkah sebagai berikut:
- Ketikkan **Firman** ke dalam sel **A7**.
 - Ketikkan **Firman1234** ke dalam sel **B7**.
 - Ketikkan **Kasir** ke dalam sel **C7**.

7. Ketikkan User dalam Name Box pada Formula Bar kemudian klik tombol **Enter**. Apabila semuanya dijalankan dengan benar, nama user yang baru dimasukkan akan ikut terseleksi.



Gambar 1.46 Nama user yang baru ikut terseleksi.

8. Ketikkan **Tabel** dalam Name Box pada Formula Bar kemudian klik tombol **Enter**. Jika semuanya dijalankan dengan benar, record data yang baru dimasukkan akan ikut terseleksi.

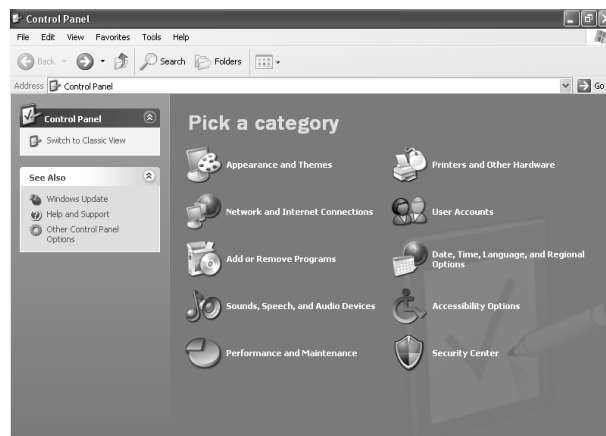


Gambar 1.47 Record yang baru dimasukkan ikut terseleksi.

1.16 REGIONAL AND LANGUAGE OPTIONS

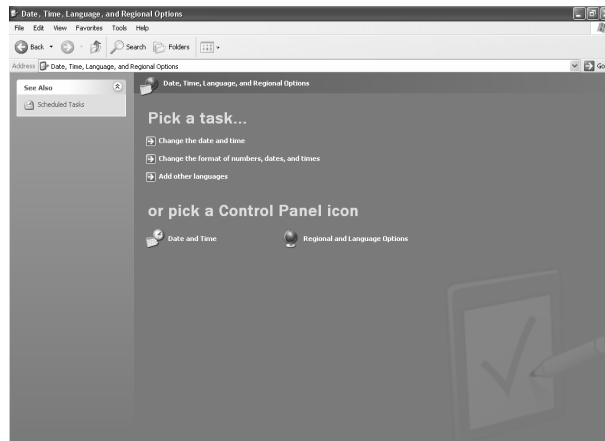
Regional and Language Options atau setting regional merupakan opsi untuk melakukan pengaturan angka, mata uang, tanggal dan waktu. Melalui pengaturan setting regional, Anda dapat mengatur penggunaan simbol koma (,) dan titik (.) untuk memisahkan nilai desimal dan ribuan. Pengaturan penggunaan simbol koma (,) dan titik (.) nantinya juga akan berpengaruh pada penggunaan titik dua (:) dan titik koma (;) dalam penulisan formula dan fungsi Excel. Untuk mempermudah dalam pemahaman materi buku, pengaturan Regional and Language Options sebaiknya disamakan dengan penulis, yaitu **Indonesia**. Berikut contoh pengaturan Regional and Language Options pada sistem operasi Windows XP:

1. Klik tombol **Start** pada taskbar, lalu pilih menu **Control Panel**. Muncul window Control Panel seperti pada Gambar 1.48.



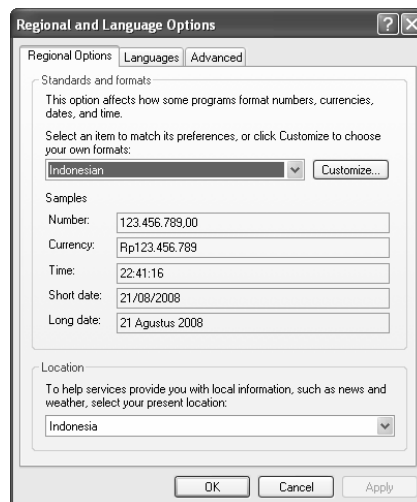
Gambar 1.48 Window Control Panel.

2. Klik ikon **Date, Time, Language, and Regional Options** pada window Control Panel. Muncul window Date, Time, Language, and Regional Options.



Gambar 1.49 Window Date, Time, Language, and Regional Options.

3. Klik ikon **Regional and Language Options**. Muncul kotak dialog Regional and Language Options.

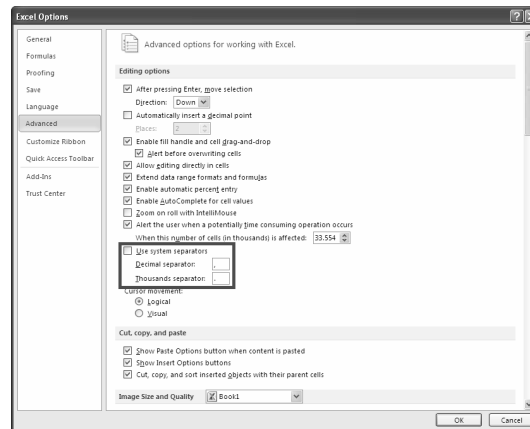


Gambar 1.50 Kotak dialog Regional and Language Options.

4. Pilih tab **Regional Options**. Pada pilihan Standards and formats, pilih **Indonesian** kemudian pilih **Indonesia** pada pilihan Location. Tekan tombol **OK**.

Selain melalui setting regional komputer, Anda juga dapat mengatur penggunaan simbol koma (,) dan titik (.) untuk memisahkan nilai desimal dan ribuan menggunakan kotak dialog Excel Options.

1. Klik tab **File** kemudian pilih menu **Options**. Muncul kotak dialog Excel Options.
2. Pilih opsi **Advanced**. Hilangkan tanda centang pada pilihan Use system separators.
3. Pada kotak isian Decimal separator, ketikkan koma (,) sebagai simbol pemisah desimal. Ketikkan titik (.) sebagai simbol pemisah ribuan pada kotak isian Thousands separator. Jika sudah, klik tombol **OK**.



Gambar 1.51 Mengubah penggunaan simbol desimal dan ribuan.